WebTransport

IETF 105 dispatch

Bidirectional Communication on the Web

	Client-Server	**Peer-to-peer**
Reliable and ordered	WebSocket	RtcDataChannel
Reliable but unordered	?	
Unreliable and unordered		

Bidirectional Communication on the Web (current)

	Client-Server	**Peer-to-peer**
Reliable and ordered	WebSocket	RtcDataChannel
Reliable but unordered	RtcDataChannel but with ICE-lite (web developers need to generate SDP provision certificates to both sides out of band themselves)	
Unreliable and unordered		

Bidirectional Communication on the Web (proposed)

	Client-Server	**Peer-to-peer**
Reliable and ordered	WebSocket (also WebTransport!)	RtcDataChannel
Reliable but unordered	WebTransport	
Unreliable and unordered		

Target applications

Anything that wants one of the following:

- “WebSockets for UDP”
- “WebSockets without head-of-line blocking”

We’ve reached out to a wide range of web developers, and there is plenty of interest in this in following domains:

- Gaming
- Live streaming

What’s WebTransport?

A common framework for exposing client-server transport protocols on the Web.

Primitives provided:

- Streams
- Datagrams

Security features required:

- End-to-end encryption (TLS)
- Origin checks
- Connection liveness confirmation

What transports are in WebTransport?

QuicTransport

- A dedicated connection, just like traditional WebSocket
- Minimal code required on top of QUIC as-is

Http3Transport

- User-created arbitrary streams with an existing HTTP/3 connection
- Fits well into traditional reverse-proxy HTTP architecture

FallbackTransport

- Potential TCP-based fallback that can be used when only TCP is available

Comparison: QuicTransport vs RtcDataChannel

	RtcDataChannel	QuicTransport
Connection model	P2P (via ICE)	Direct
Transport protocol	SCTP	QUIC
Trust model	Mutual TLS with certificate fingerprint exchanged out-of-band	Web PKI
Consent to send	Via ICE	QUIC with ALPN
Objects	Streams of messages	Streams, datagrams
Large message support	Poor (blocks the channel without NDATA support)	Just works

Comparison: QuicTransport vs WebSocket

	WebSocket	QuicTransport
Head-of-line blocking	Always	Only inside same stream
Partial reliability	None	Datagrams, cancellable streams
Trust model	TLS, Origin header	TLS, Origin header
Preventing cross-protocol attacks	SHA-1 based handshake	ALPN
Preventing middlebox confusion	XOR-based masking scheme	n/a (always encrypted)
Authentication features	Cookies	None (up to application)

Discussion time

WebTransport side-meeting

Tuesday, July 23

15:20 ~ 16:50

Room: C2 (21st Floor)

Links

- Drafts:
  - Overview
  - QUIC Transport
  - HTTP/3 Transport
  - API (WICG)
- dispatch@ discussion
- WICG discussion (a lot of developer feedback here!)